Institute of Paper Science and Technology
Atlanta, Georgia

IPST Technical Paper Series Number 926

Review, Developments and Pulp and Paper Research Applications of Data Reduction with Neural Networks

S. Karrila and S. Rezak

January 2002

Submitted to
Proceedings of Paper Summit 2002
Atlanta, GA
March 4-6, 2002

INSTITUTE OF PAPER SCIENCE AND TECHNOLOGY
PURPOSE AND MISSIONS

The Institute of Paper Science and Technology is an independent graduate school, research organization, and information center for science and technology mainly concerned with manufacture and uses of pulp, paper, paperboard, and other forest products and byproducts. Established in 1929 as the Institute of Paper Chemistry, the Institute provides research and information services to the wood, fiber, and allied industries in a unique partnership between education and business. The Institute is supported by 51 member companies. The purpose of the Institute is fulfilled through four missions, which are:

- to provide a multidisciplinary graduate education to students who advance the science and technology of the industry and who rise into leadership positions within the industry;
- to conduct and foster research that creates knowledge to satisfy the technological needs of the industry;
- to provide the information, expertise, and interactive learning that enable customers to improve job knowledge and business performance;
- to aggressively seek out technological opportunities and facilitate the transfer and implementation of those technologies in collaboration with industry partners.

ACCREDITATION

The Institute of Paper Science and Technology is accredited by the Commission on Colleges of the Southern Association of Colleges and Schools to award the Master of Science and Doctor of Philosophy degrees.

NOTICE AND DISCLAIMER

The Institute of Paper Science and Technology (IPST) has provided a high standard of professional service and has put forth its best efforts within the time and funds available for this project. The information and conclusions are advisory and are intended only for internal use by any company who may receive this report. Each company must decide for itself the best approach to solving any problems it may have and how, or whether, this reported information should be considered in its approach.

IPST does not recommend particular products, procedures, materials, or service. These are included only in the interest of completeness within a laboratory context and budgetary constraint. Actual products, materials, and services used may differ and are peculiar to the operations of each company.

In no event shall IPST or its employees and agents have any obligation or liability for damages including, but not limited to, consequential damages arising out of or in connection with any company's use of or inability to use the reported information. IPST provides no warranty or guaranty of results.

REVIEW, DEVELOPMENTS AND PULP AND PAPER RESEARCH APPLICATIONS OF DATA REDUCTION WITH NEURAL NETWORKS

Seppo Karrila
Associate Professor, IPST

Sheila Rezak
Graduate Student, IPST

ABSTRACT

Neural network modeling is often perceived as a "black box" approach, producing input-output relationships that are difficult to interpret. A further typical misconception is that their application is appropriate only with "thousands of data records".
This paper discusses neural networks applied as effective optimization tools that enable nonlinear reduction of data – not entirely unlike linear factor analysis – for further inspection by visualization, statistical analysis or other modeling, or traditional optimization. "Degrees of freedom" in relatively small data sets can be analyzed, with fundamental implications to optimization or control strategy. We provide a novel constructive method, the Nonlinear Factor Analysis (NLFA), based on examining a specific neural network configuration or topography, with conventional feedforward networks using standard learning paradigms.
Both simulated case studies and applications to pulp and paper related research data are presented. The dependence of handsheet properties on laboratory refining is visualized and discussed, and bubble size distributions in a bubble column are each reduced to two features from which these distributions can be recovered. It turns out that the effects of refining on properties of standard handsheets only have two degrees of freedom, even though data from six different types of laboratory beater are included – this is an example of fundamental insights that can be gained with nonlinear data reduction.

INTRODUCTION

Issue

Neural networks can be applied with significant benefits to the analysis of relatively small data sets, in a manner, which provides not only "non-parametric regression models" but also fundamental insights such as the degrees of freedom intrinsic to the data.
Only steady (or static) data is discussed, although similar methods can be extended to time-series or signal processing with well-known techniques. Often the number of data records from experiments providing static data is very limited, which further encourages the use of effective nonlinear methods to recover the most insights.
We provide a novel method that can be implemented with many standard software packages, because the networks are of the conventional feedforward type – it is the network configuration that makes the difference, not the learning paradigm. Our approach, termed Non-Linear Factor Analysis (NLFA), can be easily adopted with low-cost commercial software.

Significance

The reduction of multidimensional data to a lesser dimensionality, here called data reduction, is practiced regularly in the following contexts. In *pattern recognition* one wishes to select or extract significant features, typically to enable efficient classification. (Definitions of some terms will be discussed below, and references to further introductory material are given in that context.) In *information theory* data compression does encoding to a compact form, and recovery by decoding. *Visualization* supports human perception of significant features in the data, and typically requires reducing dimensionality to two or three. Finally, so-called *latent-variable models* search for a small set of "factors" that would explain the multidimensional patterns in a more interpretable manner.
Feature extraction can reduce modeling cost, improve modeling accuracy due to a phenomenon known as the curse of dimensionality, reduce measurement cost for decision making, and give a parsimonious model that needs few variables or parameters.
Typical applications of models include *steady or dynamic optimization*, the latter exemplified by process control. For each of these it is imperative to know the number of degrees of freedom (DOF), in order to apply the right number of constraints or controllers. Regression models, including non-linear ones, which do NOT incorporate

examination and use of the DOF, have "forgotten" the constraints shown by the data and thereby have lost very essential information. Then it is possible to find false optima, find inputs that according to the model should produce outputs which are not physically realizable, or without warning extrapolate outside the input data on which the model was based. The methods provided here can analyze the DOF shown by data.
Linear principal component analysis (PCA), which can be extended by factor analysis, is a well-known method of data reduction. The inherent linearity puts PCA-based methods in a well-defined setting, but also hampers the application to data with non-linear functional relationships between the variables. Since non-linearity is quite typical, unless only small variations around a set-point occur due to good process regulation, non-linear methods such as those presented here are needed for best results – linear analysis will not reduce nonlinear data to a minimal dimensionality, so it will overestimate the DOF.
To illustrate the point, consider a set of steel balls, each with a different diameter. The diameter can be selected to represent the single degree of freedom, determining the measured values: diameter, area, volume and weight of the ball (4 measured variables). Linear factor analysis will detect that the volume and weight are essentially the same variable, but there will be no further linear relationships. Linear factor analysis would then find three degrees of freedom, while in fact there is only one.

Approach

In this section we review the main concepts pertaining to data reduction and neural networks (NN). Some tools for data reduction are discussed with main emphasis on NN approaches. Then the NLFA network architecture is introduced, providing the basis for numerical exploration with simulated as well as real data.
Review of concepts and published data reduction methods. Jain and Mao provide a short tutorial on artificial neural networks in [1]. Jain *et al.* have also published an extensive review of statistical pattern recognition [2]. The interested reader is referred to these publications for a more extensive discussion than space would allow here.
The (superficial) dimensionality of data is the number of variables included in each data record. These variables may be either redundant, due to functional constraints between some of them, or only part of the information contained in them may be relevant to the modeling task at hand.
When no distinction is made between inputs and outputs, searching for patterns in the data is unsupervised. Supervised reduction of input data is possible when there are output values for a potential model, such that guide feature extraction from the input variables. A special case of feature extraction is feature selection; the extracted features are just a subset of the original variables.
The curse of dimensionality means that having more information in the form of added input variables may actually reduce modeling accuracy, when a limited number of data records (experiments) are available. An n^{th} order polynomial in d variables has $\binom{n+d}{d}$ parameters; this number grows with d like d^n – for example, a third order polynomial in 10 variables has 286 parameters. For regression analysis with inaccurate data we would need many more observations than there are parameters to fit, so the requirements on amount of data – in a nonlinear setting – grow very quickly with the number of variables. This is a pedestrian and perhaps original view of the curse, more fundamental views can be found in [2].
The minimum number of features in unsupervised data reduction, sufficient to recover all the variables without significant information loss, is the intrinsic dimensionality of the data. A more detailed discussion of this concept, as well as ways to estimate the value without constructing a mapping to reduced variables, is given in [3]. We provide a constructive method that is implemented relatively easily, and allows testing for intrinsic dimensionality, so the estimates will not be used here.
A neural network (NN) is vaguely defined as a massively parallel computing system consisting of a large number of simple processors with many interconnections. In practice the simple processors – nodes – most often only exist conceptually within a software program.
The most common type of NN is a multilayer perceptron, a feedforward network in which the nodes can be organized in successive layers, and the interconnections only feed values from one layer to the next one. The nomenclature varies in regard to how the layers should be counted; the first (input layer) and the last (output layer) may or may not be included in the count. To be explicit one should state the number of hidden layers, excluding the input and output layers. The nodes each get a linear combination (weighted sum) of the outputs from the previous layer supplemented by a bias (a constant term included in the linear combination, imagine an extra node in the previous layer which always outputs value 1), determined by the connection weights. This value is converted to node output by an activation function, which is typically linear or sigmoidal (monotonic, differentiable, with finite

asymptotes). Sigmoidal functions provide the capability to represent nonlinear relations; the logistic function is the most common of these and is exclusively used as the sigmoidal activation in the calculations reported here.
It has been shown that just one hidden layer with sigmoidal nodes has a universal approximation property; a smooth function over a bounded closed interval (possibly multidimensional) can be represented with arbitrary precision by a NN – the precision gets better as the number of hidden nodes increases. This representation is more efficient than a truncated multivariable power series, in terms of the rate of decay of error with number of model parameters, and the error can be given an upper bound for any approximated function that satisfies suitable smoothness conditions [4]. A NN with one hidden layer also trains relatively fast, with less difficulty in finding the optimal parameters than for example a 2 hidden layer NN has [5], and so is the archetypical implementation of a nonlinear mapping.
A NN finds its weights by adapting them to a training set (part of the available data records), typically minimizing the least squares error criterion. Numerically this adaptation is iterative, and the number of times the set is passed through iterations is called epochs. To make sure that the "curve fit" generated does not just fit the training data, but instead has good generalization for other consistent data, another part of the data records is held out so it does not affect training of the weight parameters. This validation set is compared at select epochs with the model, and training is typically interrupted when the validation error starts growing (while the training error still would improve). Since the validation set does affect the final network parameters through this stopping mechanism, sometimes also a third set called test set is held out, to test the generalization on data which in no way has affected the network parameters. If there is little data available, the test set is the first to go – in the work reported here we have not used a test set. Once the network has been trained with satisfactory results, it provides a fixed numerical algorithm that maps inputs to outputs – it can be used like any computational model that provides estimates based on some regressors (input variable values). We concentrate on this type of regression problems, and will not discuss in detail networks that are intended to perform classification (assigning category labels to data records).
As an analogy that helps understand generalization, we can fit a high-degree polynomial accurately to our observations, but it will oscillate wildly between the data points – it is better to interrupt the increase of polynomial degree at some earlier point and tolerate a larger error sum; the mechanism using a "validation set" would be called using a "holdout sample" by a statistician. An accessible summary of NN terminology is given in [6].
Principal component analysis (PCA) finds a linear submanifold of desired dimensionality, which contains maximal fraction of the variation in multidimensional data. A linear submanifold is a line, a plane, or a hyperplane. Mathematically PCA is essentially the same thing as singular value decomposition (SVD, also known as polar decomposition) and the numerical algorithms are well established.
While for a variable pair (x,y) regression analysis would presume that one of these variables is given accurately and the other is an observation having normally distributed errors, PCA is symmetric and minimizes the square sum of errors in normally projecting the observations to a straight line. Thus PCA is appropriate in dealing with multiple measured variables.
PCA constructs a new coordinate system that spans the submanifold, and vectors express these orthogonal axis directions with components known as factor loadings. The projections of data points on these new coordinate axes are known as scores.
PCA results can be post processed by factor analysis, which rotates the new coordinate axes in order to, for example, create such factor loadings that each score is referred back to a small subset of the original variables. Other variants of factor analysis exist, including such that create oblique coordinate systems.
PCA is an *unsupervised* method of data reduction, which only removes linear redundancies in the data. For example data with records of type (x, x^2) cannot be linearly reduced from 2-D, unless the variations in x are small. However, it is clear that this pair has only one DOF; non-linearity causes PCA to overestimate the DOF in data. Further, if the data reduction is aimed to do feature extraction and aid in generating an input-output (I/O) model, the largest variations in inputs may not be the most significant for the outputs; filtering the data through PCA may remove crucial information needed for modeling. This problem can only be removed by performing reduction of input data in a supervised manner.
Projection pursuit [7] tries to go around the problem mentioned, by seeking linear projections that have "interesting properties" in some sense – for example display clustering of data. Still, this method is both unsupervised and linear by its nature.
Before proceeding to the neural network approaches, we discuss non-linear data reduction in general.
The need for nonlinear methods has been well recognized, and statistical methods for *unsupervised* data reduction include kernel PCA, multidimensional scaling (MDS), and specifically Sammon's mapping. These methods are reviewed in [2].
Kernel PCA tries to bring in non-linearity to PCA, by first adding variables that are nonlinear transformations of the original ones. With no prior knowledge of what transforming functions should be used to fit the data well, instead

computational efficiency is sought which leads to a family of so-called kernel functions – hence the name kernel PCA.

Sammon's mapping is a special case of MDS, which strives to represent the data in two dimensions so that the distances between data points would be somewhat preserved. Such visualization can help in clustering or other human interpretation of the structure or patterns in the data.

Some variants of regression analysis can be considered to perform *nonlinear supervised reduction* of the input data. However, different users will in general get different results, as the selection of appropriate functions may require creativity. Generalized linear models first form nonlinear functions of the original variables, and use these as linear regressors to explain the output variable(s).

The methods of principal curves and principal surfaces are discussed in [8]; these are non-parametric iterative methods whose convergence properties are not well known.

Now we turn to the neural network approaches to nonlinear data reduction.

The feedforward neural network processes a data record sequentially, layer by layer in the forward direction. *Viewing the outputs from some layer as an encoded vector, the remaining layers decode this to an output vector.* In this sense, any intermediate layer provides an encoding of output data; and the layers after this one have learned a decoding algorithm.

In an auto-associative neural network (AANN) the same data vectors act as both inputs and targeted outputs of the network. A constriction or bottleneck in the network, in the sense of some hidden layer having only few nodes, provides a low-dimensional encoding for each data record in turn. Such bottleneck AANN, when successfully trained, can be split in the encoder and decoder parts – original data vectors are encoded to reduced dimension, and original data is approximately recovered by decoding.

If the bottleneck layer is the last hidden layer, the outputs will be linear combinations of the encoded values shifted by bias terms. The bottleneck shape means there are more outputs than components in the encoded data vector, so the outputs must be rank deficient – they must live on a linear submanifold. Indeed it has been proven over a decade ago, both theoretically and experimentally, that with linear activation functions in the output layer, linear PCA can be performed with a one-hidden-layer network. Various special learning paradigms have been introduced to for example ensure that the encoding corresponds to orthogonal new coordinate axes. However, for off-line analysis of static data, the capability of performing PCA with a NN has little benefits – in an on-line implementation the iterative training and adaptivity of NN is an asset though. We can divide this type of research roughly into two categories: examination of AANN properties with conventional learning paradigms, and novel learning paradigms. The latter requires special programming to implement – we will concentrate on the former methods that can be implemented with little effort using conventional software.

The same conceptual basis as for NN PCA can be used for nonlinear encoding and decoding mappings. We know that one hidden layer provides either one of these mappings with universal nonlinear approximation capability. On combining the nonlinear one-hidden layer networks so that the output of encoder becomes the input of decoder, we end up with 3 hidden layers, the middle one being the bottleneck with low number of nodes. If successful in training, the network learns to encode and decode data – this is the whole point of the concept. *The NN has the capability to learn nonlinear mappings autonomously, based on parameter optimization with some learning paradigm, and this automates generating the encoding and decoding mappings from given data.* The user needs to experiment with this constructive method to find a satisfactory number of nodes for each hidden layer. While there are information theoretic principles that can aid in this process, the conventional method of using a validation data set appears to be a practical and reliable method to avoid over-training (i.e., ensure ability to generalize) also with AANN methods.

The 3 hidden layer bottleneck AANN is known as Kramer's NLPCA network, which stands for Non-Linear Principal Component Analysis. The central bottleneck layer can have linear activation functions, so that only two of the layers are sigmoidal – this reduces the compounding of non-linearity and may speed up network training.

The NLPCA method is a global method when applied as such – it tries to deal with the whole training data set at once. (There have been attempts to create algorithms that take a local approach to dimension reduction – again these are not directly suited for standard commercially available NN software.) This global approach, while it keeps implementation and application straightforward, also limits the applicability as follows. Because both encoding and decoding are continuous mappings in NLPCA, *no discontinuities* in the data can be handled, a curve or surface (submanifold) *must not self-intersect* (the encoding mapping can not jump to one branch or another at the intersection point), and *"projection" of new data into the submanifold may behave erratically* close to "ambiguity points" [9]. By projection we mean, that new data fed into the encoder will be reduced numerically to coordinates describing the submanifold, and the decoder will give a corresponding point in the submanifold – this maps the new data point to another point that is similarly constrained as the training data, and this mapping is the "projection". The mapping is approximately idempotent (i.e., $P = P^2$ because points already on the submanifold are projected

approximately to themselves), but it is neither linear nor orthogonal – it is not the kind of projection we learn in linear algebra.

Another noteworthy issue is the *non-uniqueness* of the encoding-decoding pair. We can add an invertible function to post-process the encoding, and pre-process with the inverse before decoding – and the new pair of encoding and decoding is no better or worse than the original. The reduced data values must be considered purely artificial mathematical constructs. In analogy to PCA some authors call these values the scores – in this article we'll call them state variables. This is somewhat analogous to thermodynamics where any pair of state variables will fully define the state of a pure substance, and constructs such as S, G, H, U and F which can not be directly measured are still useful.

A relatively popular and accessible NN paradigm, implemented in many software packages, is Kohonen's self-organizing map (SOM). DeBacker has compared linear PCA, MDS, Sammon's map, SOM, and AANN for classification performance – the result was somewhat expected in that all the nonlinear methods outperform PCA. The NN based methods (SOM and AANN) were most appropriate when there is enough data relative to number of variables, while the first two nonlinear methods perform classification better if data is scarce [10]. Our main interest is not in classification, but in regression type problems.

To end this review, we'll briefly discuss some developments that currently are not easily accessible (without significant own coding work). Sammon's mapping in its conventional form does not allow injection of new data without redoing the whole calculation. A NN method has been developed to perform this mapping, removing this problem [11]. This network method was compared with other classification schemes, and with these test cases a NN method known as NDA was superior to others. NDA is performed with a (bottleneck) network using 2 hidden layers for nonlinear encoding and linear decoding [12, 13] with supervised training.

An introduction to the NLFA method. It is typical in the literature that whenever the decoding is nonlinear, so is the encoding. However, linear decoding is often combined with nonlinear encoding, as is done in NDA.

We can denote the type of encoding-decoding pairs by L for "linear" and N for "nonlinear", so that PCA is of type L-L, while NLPCA is of type NL-NL. NDA is of type NL-L, but then this is not auto-associative. Auto-association with only one hidden layer has been studied in the dissertation of Japkowicz, but also she includes only types L-L, NL-NL, and NL-L [14]. As discussed earlier, the last of these would imply that the data could be equally reduced by linear PCA, so only the NL-NL type is interesting in trying to surpass classical linear methods with bottleneck type NN methods.

It then seems that type L-NL encoding-decoding pairs are avoided intentionally. The reason for *unsupervised* learning (auto-association) may be thinking in the following manner: "the decoding is the inverse mapping of encoding, if the encoding is linear its inverse is also linear, so we end up with L-L type anyway". A counterexample to such thinking is the redundant pair (x, f(x)), which can linearly be mapped to x, from which the original pair is nonlinearly recovered. *The point is that when superficial dimensionality is not conserved, the inverse of a linear one-to-one map may be nonlinear.*

For hetero-associative or *supervised* learning the L-NL type is actually frequently practiced, but in a special form. Feature selection again is a projection of the input vectors, a special case of linear mapping. Then the L-NL type bottleneck network can perform at least as well as feature selection based procedures, but *the linear feature extraction is done automatically* as part of the network training. While we have applied our method in a supervised manner, that case study can not be included in this publication.

We have established that a L-NL type encoding-decoding pair has useful properties, for both supervised and unsupervised data reduction. The implementation of this method is rather obvious: the first hidden layer is a bottleneck with linear activations, while the second hidden layer is sigmoidal, and no more hidden layers are necessary. The reduced input data is given by the outputs of the first hidden layer, and this data reduction can be explicitly written out as linear relations.

This new method deserves a name for the following reasons.

- Despite its simplicity, the method is non-obvious and has not been practiced
- The method has added value over auto-associative analyses since it can be used for supervised learning (reduction of input data based on outputs)
- Documenting the data reduction is easy in the form of linear coefficient vectors – this improves usability over fully nonlinear approaches such as NLPCA which produces a non-transparent NN algorithm for encoding and decoding each
- The restricted set of encoding mappings makes the results less user dependent (in some cases unique, possibly after some post-processing) compared with NLPCA, which gives mappings that may not be independently duplicated from the same data.

The name given here is **Non-Linear Factor Analysis (NLFA)**, which is appropriate because the encoding recovers oblique factors that are linearly constructed from the inputs with "factor loadings" – in fact these can be post-processed with conventional PCA or factor analysis.
In practical tests the NLFA surprisingly outperformed NLPCA (training with backpropagation gave smaller reproduction error), while it obviously will outperform PCA whenever the data contains nonlinear dependencies.
The greatest part of work in applying the NLFA method is in trying out different numbers of nodes – and retraining the net several times to ensure that the random starting values for optimization have given a "good" converged set of parameters. Once the encoding has been found and documented, it can be applied to learn the corresponding decoding with the same or compatible data. For this reason often it will be sufficient to just document the encoding – and NLFA will give this in nice linear form. The linear encoding has further benefits that will be discussed in future publications.
If the NN always were able to find the global optimum for its parameter values, then NLFA would be somewhere between PCA and NLPCA in its scope: restricting the encoding or decoding to linear functions is a subset of the case where also nonlinear mappings are allowed. (In practice a sigmoidal function is nearly linear locally, and the interconnection weights can magnify and make use of only such portion of the activation function. Thus a sigmoidal layer is capable of doing linear mappings if it finds this optimal.) However, as network complexity increases also the difficulty of finding a global optimum increases – instead network training will tend to converge to a local suboptimal minimum error. The NLFA method has lesser complexity than the NLPCA method, in which the errors need to be backpropagated further "upstream" to optimize the nonlinear hidden layer in the encoding part. This is a potential explanation why NLFA can in practice sometimes outperform the NLPCA, even though the latter in theory provides a superset of the modeling capability of NLFA.
Further discussion of the theoretical aspects of NLFA will be published elsewhere.

RESULTS

Simulated Data – Step Function

Data representing a unit step, traced at constant speed, was manually generated in a spreadsheet.

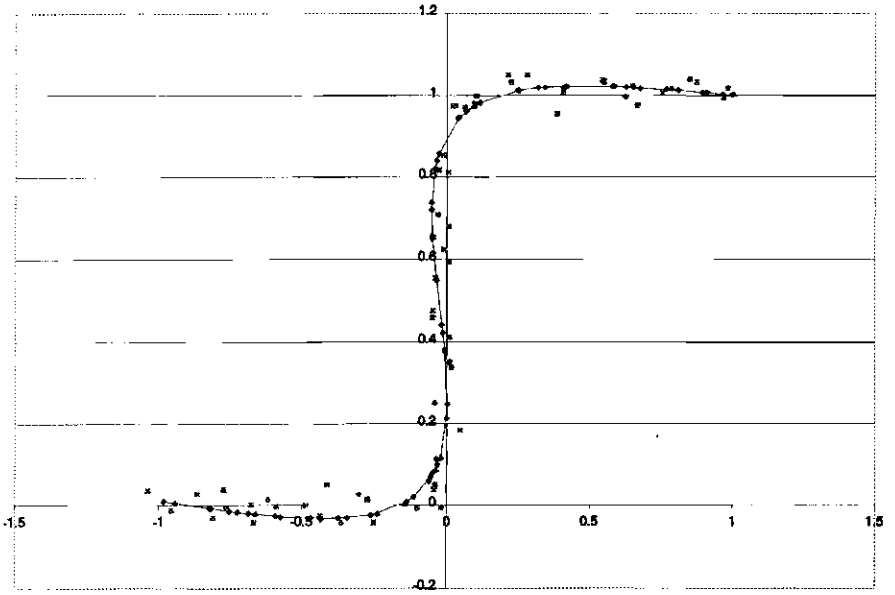

Figure 1. Two-dimensional data that does not allow linear reduction or straightforward regression is modeled with NLFA. The solid line shows the model as well as projected data points (recall that this projection is not orthogonal).

The coordinates were perturbed with jitter to simulate measurement noise, by adding the function "-.05 + .1*Rand()" which is evenly distributed in the range [-0.05, 0.05], before converting to fixed values so that the "Rand" function would not be recomputed.

The data has superficial dimensionality two, but the curve can be traced with one parameter – the intrinsic dimensionality is one. Clearly regressing y on x or vice versa is doomed to fail, the step function in this case is not a single-valued function of x or y. Also, PCA would show no opportunity to reduce dimensionality.
The NLFA method finds a one-dimensional "encoding" or state variable s, which on decoding traces a curve (x(s), y(s)). The data with jitter and the recovered curve are shown in Figure 1.

Simulated Data – Helix

A 3-D helix described by {x=cos(t), y=sin(t), z=t/5}, with t in range [0,8], was generated in a spreadsheet using step size dt=0.1. Jitter similar to the previous simulated example was added to make the 81 data points "noisy".
This represents a curve in space similar to a spread-out key ring; the range of t-values corresponds to about one and a quarter cycles around the z-axis. It is clear that the data lives on a 1-D manifold – a single parameter t (alternatively z) is needed to generate it. Projection to z-axis is then an appropriate linear 1-D encoding – on the other hand linear encodings with significant weight on x or y will not be feasible, because the corresponding constant-value planes would multiply intersect the curve. This is a case where the NLFA will produce (nearly) user-independent results; restricting the encodings to linear allows only one solution with one DOF (minor deviations are still numerically allowed with such limited range of t-values).
The NLPCA method may generate any parameterization s(t) of the curve, where this mapping s just needs to be monotonic – there is no uniqueness for fully nonlinear data compression.
NLFA needs to create such linear projection of the data that the x and y components have negligible effect. The problem is significantly harder than the previous step function approximation; very slow learning rate and low momentum are necessary to make conventional backpropagation training algorithm work. In practice learning rate 0.01 and momentum 0.05 worked well. The sigmoidal layer had 10 nodes. On a 1GHz laptop with other programs open in the background, the training time is about one minute.
It should be clear that this type of problem is not manageable with a linear method. Further, the problem is nearly pathological, exhibiting extreme non-linearity combined with significant redundancy (two out of three variables are redundant).

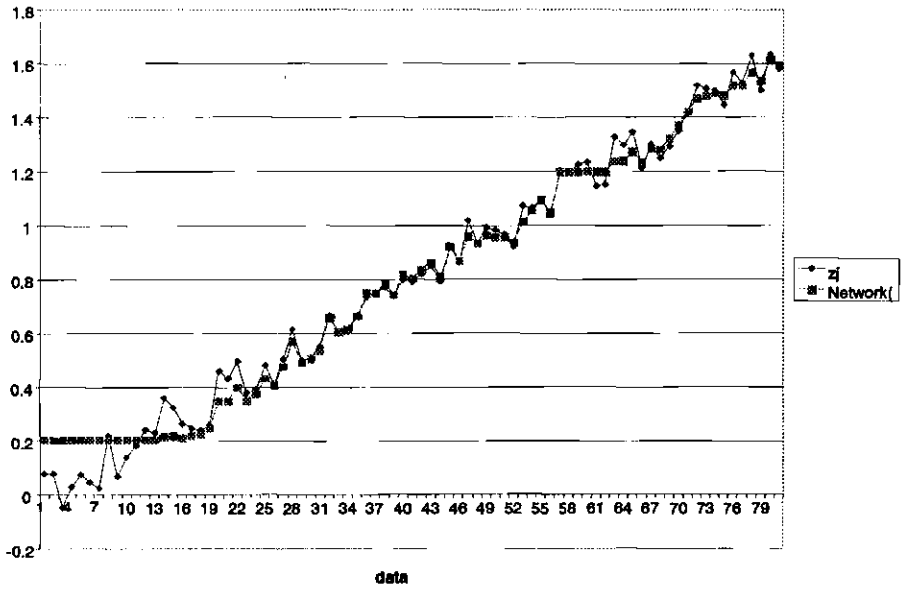

Figure 2. Axial z-coordinate of the helical coil with jitter is shown with the network output generated from one state variable.

The data (with jitter) is displayed together with the NN estimates in Figures 2, 3, and 4. Note the initial large error in the estimates. It turns out that the linear activation function of the bottleneck node (a layer with a single node to recover the one state variable) is truncated; it is piecewise linear and "saturates" with too large or small values – this is a quirk of the commercial program used. To get rid of this problem a sigmoidal activation function was tried; this approach works as illustrated in Figure 5.

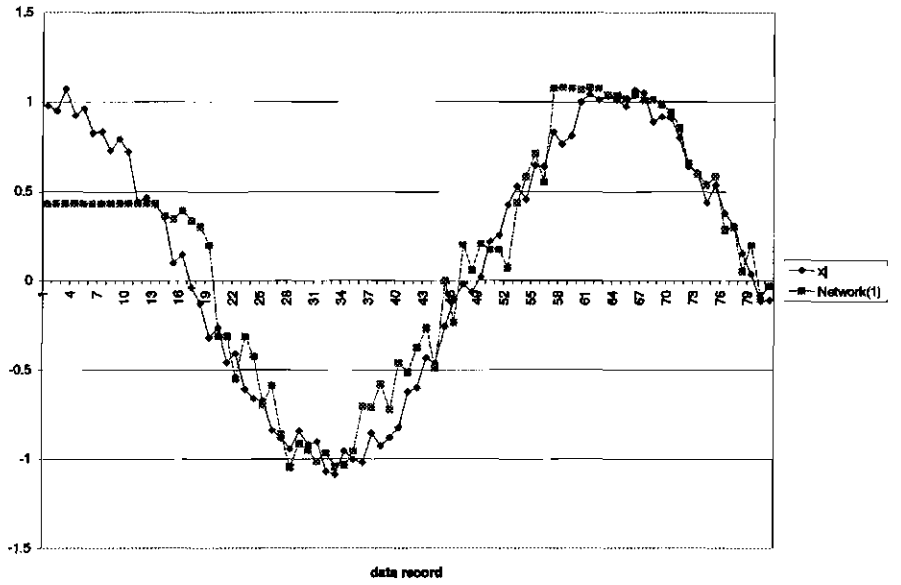

Figure 3. Lateral x-coordinate of the helical coil with jitter is shown with the network output generated from one state variable.

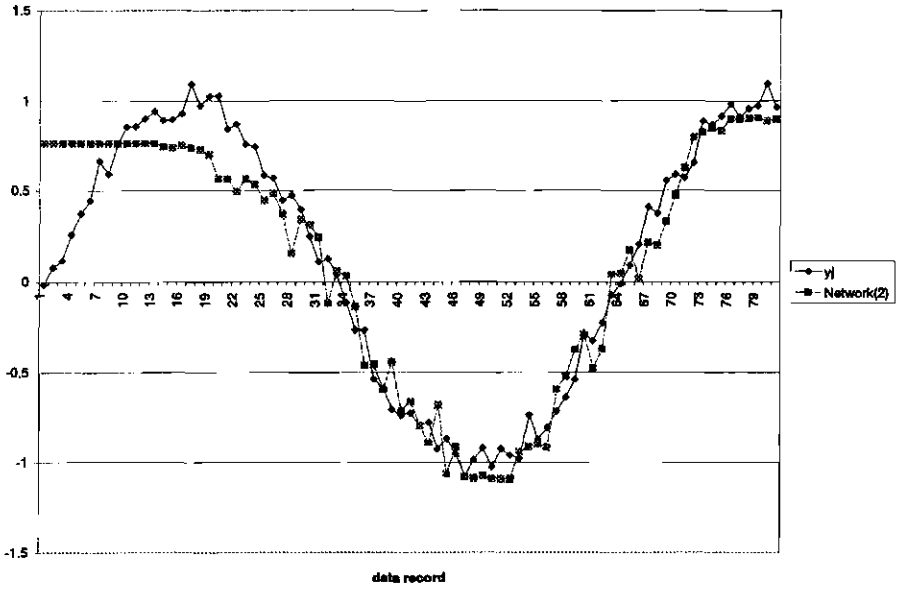

Figure 4. Lateral y-coordinate of the helical coil with jitter is shown with the network output generated from one state variable.

The use of a sigmoidal activation does not prevent from extracting the linear mappings feeding into the bottleneck layer – also computing time is unaffected. Instead of recording the layer output, it is now necessary to revert to the interconnection weights feeding this layer. The difficulty encountered illustrates that also commercial software may have surprising features or actual bugs – it does not in any manner invalidate the generic NLFA approach.

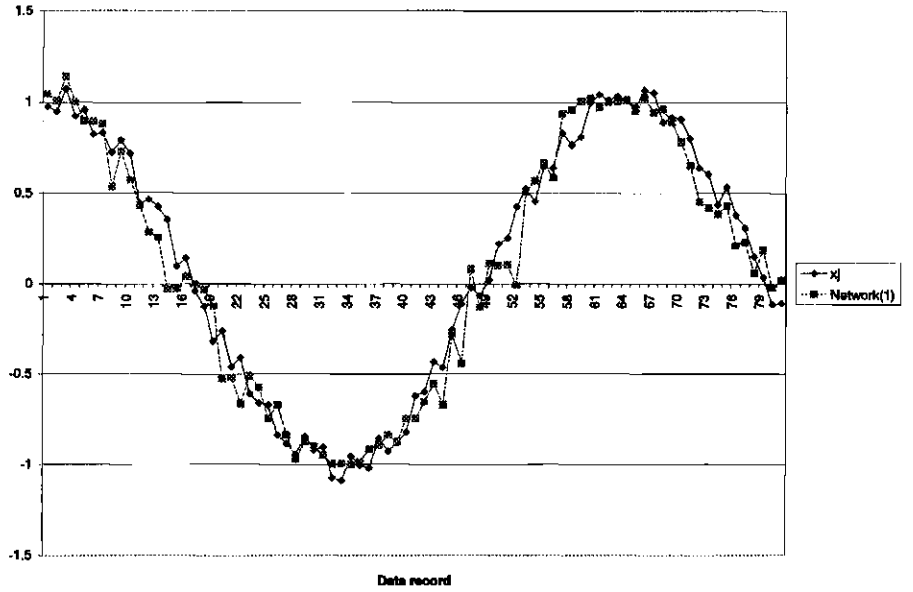

Figure 5. Lateral x-coordinate of the helical coil with jitter is shown with the network output generated from one state variable. The activation function for the single node in the bottleneck layer is sigmoidal.

This simulated case showed how the NLFA method can perform optimization-based automatic feature extraction even in an auto-associative setting. If the results were post-processed manually feature selection would be indicated by the low weighing of x and y inputs. Of course one can now directly construct a very good NN model with just z as input. However, the NLFA method would work equally well even if the coil data were rotated in 3-D space, which makes simple feature selection a useless option for data reduction.

Laboratory Refining Data

Howard, Poole and Page [15] applied linear factor analysis to a published collection of data [16], of which we only inspect a subset. Up to the point of rotating the principal components the results of such analysis are user independent. Further background on linear factor analysis (and principal component analysis) can be found in the references of their publication.

They found 3 major factors and gave an interpretation for each – with any method (including NLFA) one could dispute where the cutoff in required reproduction error should be; equally well four or five factors could have been retained from PCA, but the available interpretations happened to have a good match with only three factors.

Our data comes from Appendix I, Table 16, of the same reference. These data are for a bleached sulphite pulp labeled "extra strong (green)". Six different types of beater were applied to the same pulp for various periods of time, by different laboratories – due to a duplication of the Lampen mill, there are seven "beating curves".

We model 10 variables with the NLFA method. These are freeness, breaking length, Mullen burst factor, tear factor, Bekk porosity (transformed with natural logarithm), Schopper fold (transformed with natural logarithm), % stretch, scattering coefficient, contrast ratio, and apparent density.

Two state variables are found to conserve the measured data within reasonable reproduction error. The lowest R^2 – value 0.88 is found on predicting stretch, mainly due to inaccuracy in the measurement of this variable. With the exception of natural logarithm of porosity (value 0.96) and tear (0.97), the remaining variables have R^2 –values around 0.99.

Only 8 sigmoidal nodes are needed in the second hidden layer – the first has two linear nodes as stated above. There are 40 data records, and 20% of these were held as validation data during training.

To illustrate the worst case fit the measured and estimated stretch values (in %) are shown in Figure 6.

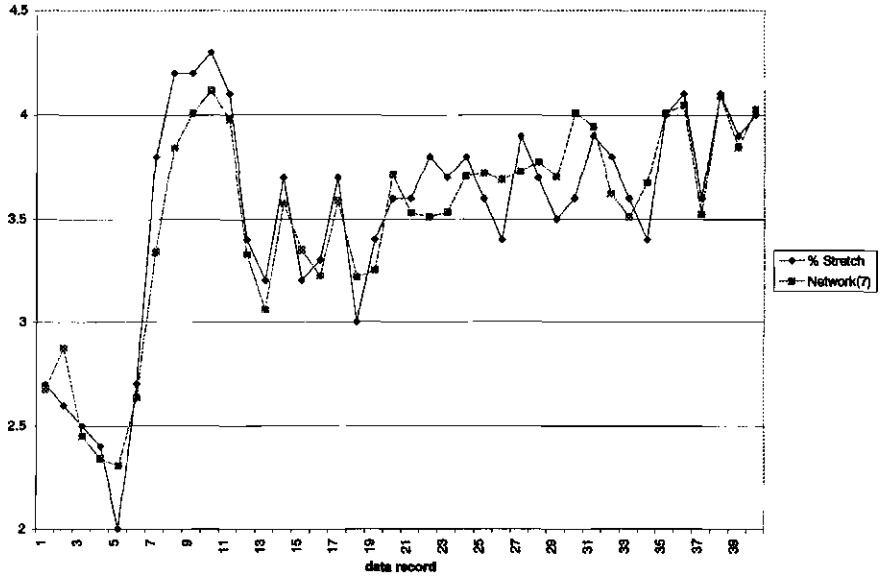

Figure 6. The error in reconstructing stretch % from 2 state variables is illustrated by curves showing the measured and recovered values in the sequence of data records. The R^2-value is about 0.88, while for other variables this is better than 0.96 – this is the worst case of the ten measured properties of standard handsheets.

The result is of fundamental nature: from standard handsheet properties we can only observe two independent effects of laboratory beating. Note that a conclusion along the lines "laboratory beating effects on pulp have only two degrees of freedom" is NOT justified – the observation method is an essential part of the process that generated the data, and all conclusions must take this into account. As a speculation, if the sheets had been calendered to various degrees, further DOF's in the influences of laboratory beating could have surfaced from the data.
While details are not reported here, data for another pulp was combined with the data above and processed similarly – only two DOF's were sufficient for the combined data still. This indicates that the result is not just a coincidence but has some general validity.
If this were a process to be optimized, we could only give target values for two handsheet quality variables; these determine the remaining ones. Further, once a model is established we could select two measurements that most accurately determine the state variables (within some operating region) and avoid the cost of the remaining redundant measurements.

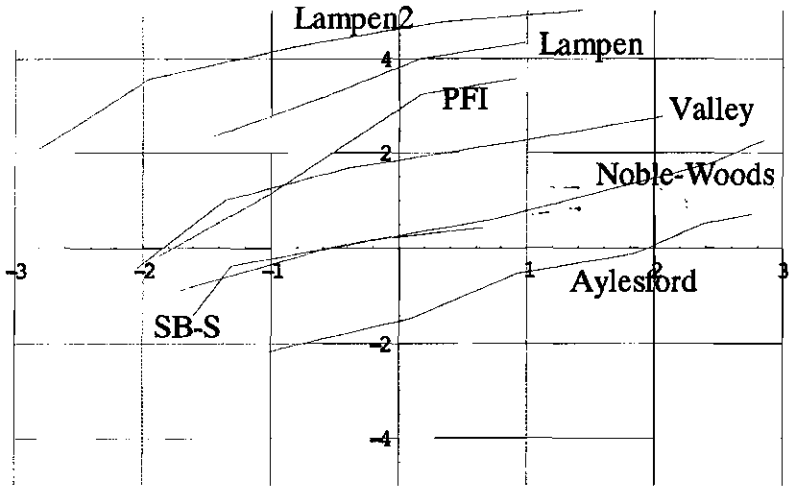

Figure 7. Beating curves of different types of laboratory beater, shown in the state variable coordinate system. SB-S indicates Seyboldt-Banning, the other codes used are self-explanatory.

The beating curves for various types of beater are shown in Figure 7. Aylesford and Lampen mill are at opposite ends of the spectrum, in terms of the type of effect on pulp, while the Valley and PFI are fairly similar and in the middle. The two Lampen mills included in the multi-laboratory study perform slightly differently.
The variables dependent on the two state variables, i.e., all the measured handsheet properties, can be plotted as contours and overlaid on these beating curves. We only show a few of the ten variables in the plots below.

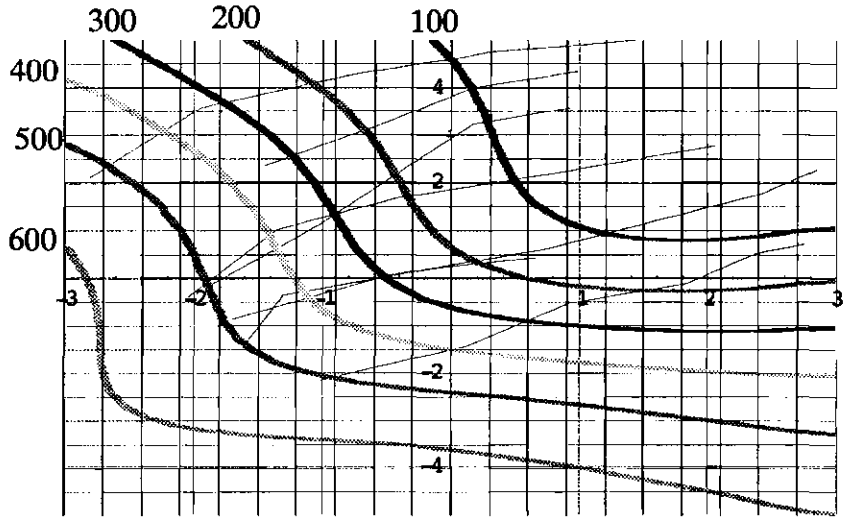

Figure 8. The development of CSF is shown along the refining curves of Figure 7 by the contours. The non-linearity of mapping from state variables (coordinates used in plotting) to CSF is apparent by the curvature of the contours.

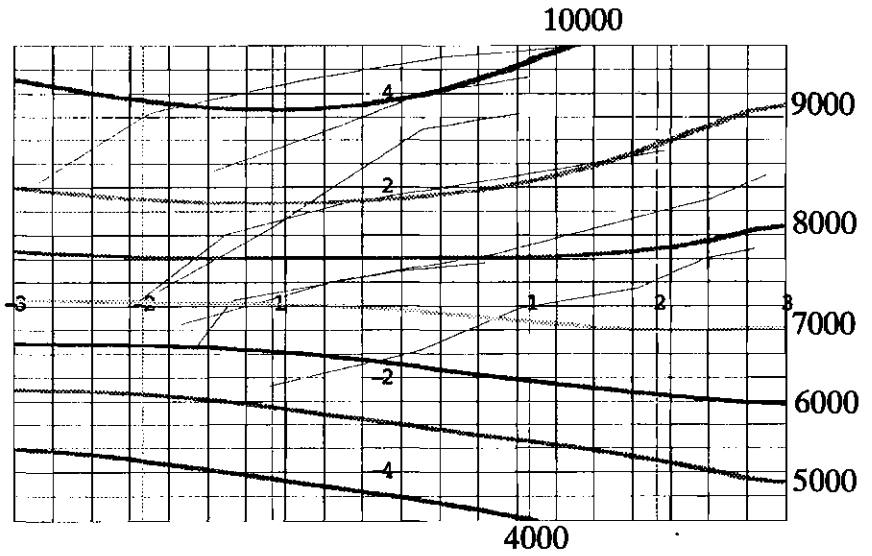

Figure 9. Breaking length shown as contours overlaid on the refining curves.

These visualizations display in a tangible fashion the effects of laboratory refining. Similar plots could be used to "manually" optimize a process, or to determine the state variable combination from any two measured values (the accuracy will depend on which measurement pair is chosen – the selection should be based on a sensitivity analysis, which can be performed with help from plots such as above).
The neural network training time with this problem, per individual network, is of the order 15 seconds. However, significant processing time may be required to vary the number of nodes in the hidden layers and try out the effects, inspecting the results both numerically (R^2-values and other goodness-of-fit characteristics) and graphically. With these small data sets, data preprocessing is not typically a major issue once the data are in electronic format.

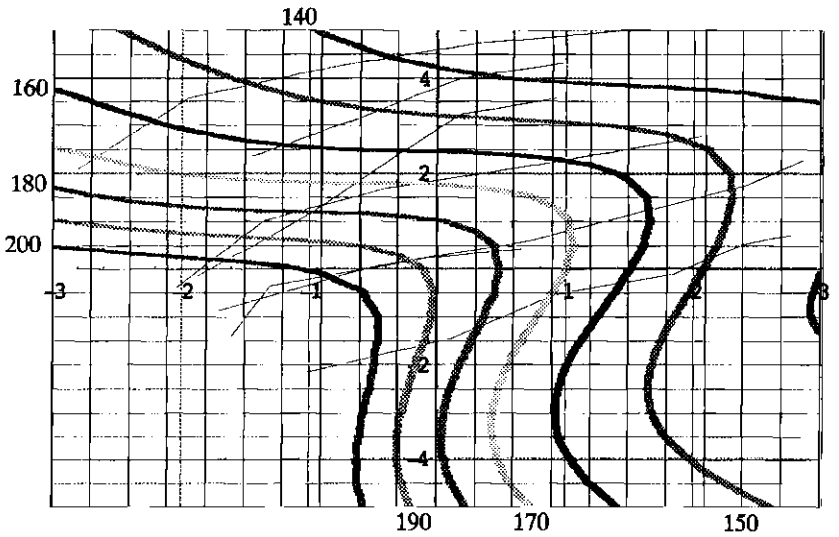

Figure 10. Tear factor shown as contours overlaid on the refining curves.

Bubble-Size Distributions in Pulp Slurry

Bubble size distributions were determined in a flow column with pulp slurry, using X-ray imaging and image analysis. The manipulated variables affecting this size distribution were the slurry and gas flow rates, slurry consistency, initial mixing intensity, and imaging position along the column. 55 data records were collected.
The size distribution measurement produces size categories with upper size limit for diameter equal to 1, 2, ..., 16, and ∞ mm (the last category includes everything over 16mm). A cumulative distribution function (CDF) will reach value 1 for the last category, so this value has no information content – a vector with 16 components describes the normalized cumulative distribution measured.
For size distributions of particulate materials there are several well-known trial functions; we tried Rosin-Rammler-Bennett, Gates-Gaudin-Schumann, Gaudin-Meloy, lognormal, and an improvised modified lognormal distribution. The modified lognormal distribution provided excellent performance in fitting the measured CDFs. This distribution is given by

$$50c\left(Erf\left(\frac{\ln(\frac{x}{a})}{b}\right)+1\right)$$

where a, b, and c are parameters to be identified.
This classical approach – find function shape, fit parameters – reduces the data records to 3 parameters, so this is the maximum number of DOF. From the manipulated variables we already knew that there are less than 5 DOF.
The NLFA approach showed that there were in fact only 2 DOF in the collected data. This means that any effects on, or effects of the bubble size distribution need to be referred only to these two parameters, which significantly eases the modeling effort.
In terms of the effort, the conventional approach to data reduction took a couple of days of work. This included finding and programming the function shapes to be tried, then going through all of the data records with non-linear optimization to fit the parameters, and finally assessing the results from numeric indicators and graphics. The NN approach using NLFA was doable in about two hours, as it was directly applied to all the data records stored in a spreadsheet – and further it gave a better reduction of the data. This illustrates the laborsaving aspects of the NLFA, which is typical also in other NN-based approaches; user intervention is not needed to select and program trial functions.

CONCLUSIONS

Reduction of dimensionality can enable visualization, facilitate modeling, and by revealing the intrinsic dimensionality of data it can have implications on measurement practices, optimization, and process control. Data reduction finds functional constraints between variables that are latent in the data records by constructing a reduced set of state variables and mappings that implicitly constrain the original variables. These constraints can be invariances of fundamental nature with wide applicability – the first principles theoretical approach is not the only one on seeking such fundamental insights, the data-driven approach with laboratory data provides an option. More practically, the constraints can limit which variable combinations are possible to physically achieve, so knowledge of them is essential on optimizing a process.

The **Non-Linear Factor Analysis** or **NLFA** is a novel tool that gives access to the benefits of data reduction. A review of prior methods for non-linearly reducing the dimensionality of data shows that the NLFA method presented is novel. Its potential has been examined with simulated and real-world examples to verify that the approach works not only in principle but also in practice. The NLFA can be implemented with standard feedforward neural networks – its advantage over black box neural network modeling has been illustrated with applications to experimental data.

Benefits of the NLFA method include the following.

NLFA is more flexible than linear methods such as PCA, and will often find a tighter estimate of the DOF (or intrinsic dimensionality) of data – it reduces data more than linear methods can when non-linearity is present.

Compared with NLPCA, the NLFA is easier to train; the encoding results are expressible with linear equations that can be post-processed (for example with linear PCA) and on occasions the encodings can be independently reproduced while no such uniqueness is expected with NLPCA.

The real-world cases included illustrate how fundamental insights are found from data. From published data it was shown that laboratory refining, despite using several different types of beater, only affects standard handsheet properties through two state variables of the beaten pulp. The other case based on laboratory data showed an advantage over traditional non-linear curve fitting, both in the amount of labor and in the insights provided by the results – we expect that our approach will prove especially useful with any kind of measured distributions as a pre-processing stage that precedes other modeling.

ACKNOWLEDGEMENTS

IPST and its Member Companies have supported graduate student Sheila Rezak – her measurements of the bubble size distributions have also been supported by DOE grant DE-FC07-00ID13871, which is gratefully acknowledged.

References

1. Jain, A.K., Mao, J., "Artificial Neural Networks: A Tutorial", Computer, 29(3), March 1996, pp. 31-44 (1996)

2. Jain, A.K., Duin, R.P.W., Mao, J., "Statistical Pattern Recognition: A Review", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(1), Jan. 2000, pp. 4-37 (2000)

3. Verveer, P.J., Duin, P.W., "An evaluation of intrinsic dimensionality estimators", IEEE Trans. Pattern Analysis and Machine Intelligence, 17(1), January 1995, pp. 81-86 (1995)

4. Barron, A.R., "Universal approximation bounds for superpositions of a sigmoidal function", IEEE Trans. Information Theory, 39(3), May 1993, pp. 930-945 (1993)

5. Villiers, J. de, Barnard, E., "Backpropagation neural nets with one and two hidden layers", IEEE Trans. Neural Networks, 4(1), January 1992, pp. 136-141 (1992)

6. Stegemann, J.A., Buenfeld, N.R., "A glossary of basic neural network terminology for regression problems", Neural Comput. & Applic (1999)8:290-296 (1999)

7. Peshkin,L., "Dimensionality reduction – a primer", available from www.ai.mit.edu/~pesha/Public/papers.html

8. Carreira-Perpinan, M.A., "A review of dimension reduction techniques", Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, (January 1997)

9. Malthouse, E.C., "Limitations of nonlinear PCA as performed with generic neural networks", IEEE Trans. Neural Networks, 9(1), Jan. 1998, pp.165 –173 (1998)

10. De Backer, S., Naud, A., Scheunders, P., "Non-linear dimensionality reduction techniques for unsupervised feature extraction", Pattern Recognition Letters 19, pp. 711-720 (1998)

11. Jain, A.K.; Mao, J., "Artificial neural network for nonlinear projection of multivariate data", International Joint Conference on Neural Networks 1992 (IJCNN), Volume 3, pp. 335 -340 (1992)

12. Mao, J.; Jain, A.K., "Discriminant analysis neural networks", IEEE International Conference on Neural Networks, pp. 300 -305 vol.1 (1993)

13. Mao, J., Jain, A.K., "Artificial neural networks for feature extraction and multivariate data projection", IEEE Trans. Neural Networks, 6(2), March 1995, pp. 296-317 (1995)

14. Japkowicz, N., "Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification", Dissertation, the State University of New Jersey, New Brunswick Rutgers, 1999

15. Howard, R.C., Poole, R, and Page, D.H., "Factor analysis applied to the results of a laboratory beating investigation", J. Pulp and Paper Science, 20(5):J137 (1994)

16. Cottral, L.G., Stephansen, E., Tyden, A., Rosen, W., and Nybergh, B., "Comparative beating investigation with different laboratory beating apparatus for pulp evaluation", Proc. Tech. Sect., BPBMA, 35(3):359 (1954)